N89-24650

NONLINEAR HIERARCHICAL SUBSTRUCTURAL PARALLELISM AND COMPUTER ARCHITECTURE

Joe Padovan
The University of Akron
Akron, Ohio

Abstract

This paper investigates computer architecture in conjunction with the algorithmic structures of nonlinear finite-element analysis. To help set the stage for this goal, the development is undertaken by considering the wide-ranging needs associated with the analysis of rolling tires which possess the full range of kinematic, material and boundary condition induced nonlinearity in addition to gross and local cord-matrix material properties.

1. Introduction

With the advent of the finite-element method (FEM), the analysis of large-scale structure is finally possible. While large-scale linear finite-element simulations are relatively economical, such is not the case for nonlinear situations involving geometric, material and boundary induced nonlinearity[1-4]. There are numerous aerospace and commercial structures which require full-scale nonlinear analysis to enable their improved design. This includes such structural systems as gas turbines, space structures, aircraft structure, autos, etc. Perhaps the most commonplace of such structures is the tire, which serves as a component to a wide variety of aerospace and auto systems.

To bypass the difficulties associated with nonlinear FE analysis, significant work has been channeled into two main areas, namely:

i) The development of algorithmic improvements, element-element,[5] constrained Newton/Raphson (NR),[6] and hierarchical least squares,[7]

ii) The design of new computer architecture enabling hardware speedup, i.e., as in vector processors (Cray, Cyber 205 and true parallel machines[8,9])

In the context of such thrusts, not enough effort has been undertaken to consider how algorithmic structures might effect machine architecture or vice versa.

Based on the foregoing comments, this paper will investigate machine architecture in conjunction with algorithmic structure. To achieve this goal, the development will be undertaken by considering the wide-ranging needs associated with the analysis of tires. This approach was taken since, as will be seen in later sections, the needs of tire modeling embody essentially all the requirements of nonlinear continuum mechanics, namely[10]

i) Material nonlinearity

ii) Inelastic behavior

iii) Large deformation/strain kinematics

iv) Complex inertial fields

v) Nonlinear boundary conditions

vi) Microstructure

vii) Thermomechanical response

viii) Solid fluid interaction

All this leads to the development of what is called hierarchical substructural parallelism which enables bottom-up/top-down modeling.[11] Overall a nonlinear multilevel substructuring scheme is overviewed which enables the simplification of the data based management (DBM) of parallel-type operators while still yielding enhanced computational speeds as well as reducing core requirements.

In the sections that follow, detailed tire modeling discussions embody the diversity of needs of nonlinear simulations, various types of current machine architectures, and potentials of hierarchical substructural parallelism. Examples that define enhanced properties will also be given.

2. Shortcomings of FEM Vis-à-vis Tire Structural Analysis

Noting Figure 1, the tire possesses a very regionalized/substructural form of construction. Overall it consists of:[12]

i) Carcass plies, steel/glass/Kevlar cord-rubber composites

ii) Belt plies (same as above)

iii) Bead, bundled steel cords

iv) Thread configuration

v) Regionalized rubber types

vi) Belt edges, turnup plies

The operating environment consists of:

i) The tire-road interface which involves varying pavement textures, flexibilities and resulting frictional characteristics[13,14]

ii) The tire-rim interface

iii) The tire-rim-suspension behavior

iv) Cornering, braking and accelerating maneuvers

v) Standing, steady/transient rolling[10,13-15]

vi) Obstacle/hole envelopment roll over events[13,14]

vii) Pressurization[16,17]

As seen from Figures 2 and 3, the pressurization and subsequent loading into standing contact can lead to large deformations and associated rotations. For instance Table 1 illustrates comparisons of the deflection fields generated from linear and nonlinear FE simulations.

In this context, it follows that there are several sources of response nonlinearity, namely

i) Large deformation kinematics

ii) The road-tire-rim interfaces

iii) Bimodular behavior of cord-rubber composites in transitions from tension to compression

iv) Thermomechanical interactions

v) Material nonlinearity

vi) Local large strain levels in various regions of the tire; belt edges, bead region, and tread

vii) Dynamic impact interactions

Each of the foregoing sources of nonlinearity initiates different forms of response behavior.

For instance, from a kinematics point of view, the pressurization process causes rotations and deflections which lead to an overall stiffening of the tire. Similarly, as with Hertizian contact problems, the tire-road interface also exhibits hardening-type properties, namely, the hub force-deflection response is stiffening in character as noted in Figure 4.

In addition to the foregoing modeling difficulties, in general the tire response needs to be handled in several levels, namely

i) Cord-matrix and regionalized rubber interfaces

ii) Whole cord-rubber plies/laminae

iii) Full laminate structures, several plies as in belt and carcass laminates

iv) Full (global) structure

As one preceeds from (i)-(iv), a "bottom-up" modeling[11] approach is required wherein fine detail is handled at the lowest level while the upper level models are increasingly coarser so as to reduce overall degrees of freedom in a global model. Once the global-level model is solved what is needed is a "top-down" scheme[11] to provide proper mechanics information at the constituent level. Such an approach is necessary if proper stress and strain fields are to be captured hence enabling proper description of internal fields.

Current FE models of tires start from level (iii) and proceed to (iv). In this way, a true local-level description of mechanical fields is not possible.

3. Types of Parallelism

Multiprocessor computers fall basically into two main categories, namely

i) Vector processors (Cray, Cyber 205)

ii) True parallel processors (Flex, Goodyear)

Compared with single processor units (IBM 3084, CDC7600), vector processors enable quicker more efficient handling of matrix manipulations. This is achieved through the use of multiple processors which operate simultaneously on a succession of matrix elements. Data transfer for such operations is typically from a single common core storage.

In true parallel processors, different functions/operations are performed in separate processors. In such machines data transfer usually involves both a common core as well as individual local processor cores. For such machines very high speeds can be realized.

In the context of programming languages, vector processors typically can be programmed in enhanced versions of FORTRAN or the like. For true parallel processors, overall programming is generally achieved at two levels. At the local processor level, languages such as FORTRAN can be employed. At the total system level, machine control language MCL is usually employed.

4. Classical Solution Algorithm

The solution of large-scale FE simulations typically involves either some variant of the Newton/Raphson scheme NR, or an explicit/implicit time integration procedure. For the current demonstration purposes, the presentation will concentrate on static equation solvers. The most recent improvements for such problems fall into several categories, namely

i) Element-by-element preconditioners (Hughes et al.[5])

ii) Constrained NR procedures of Padovan and Arechaga[6]

iii) Constrained hierarchical least-squares algorithms of Padovan and Lackney[7]

Assuming large deformation kinematics along with potential material nonlinearity, the governing FE formulation takes the form[6,7]

$$\underset{\sim}{F} = \underset{\sim}{G} + \int_R [B^*]^T \underset{\sim}{S} dv \tag{1}$$

where $\underset{\sim}{S}$ is the second Piola Kirchoff stress tensor, $\underset{\sim}{F}$ is the nodal force vector and $\underset{\sim}{G}$ is the vector of body forces. Typically (1) is nonlinear and must be solved via NR schemes. After expansion into truncated Taylor series, (1) yields the following NR algorithm namely[1,2,6,7]

$$\Delta\underset{\sim}{G} + [K_i]\Delta\underset{\sim i+1}{Y} = \underset{\sim}{F}+\Delta\underset{\sim}{F} - \int_R [B_i^*]^T \underset{\sim i}{S} dv \tag{2}$$

where [K] defines the tangent stiffness matrix, that is[6,7]

$$[K_i] = \int_R [[G]^T [S_i][G] + [B_i^*]^T [D_{Ti}][B_i^*]]dv \tag{3}$$

such that $[S_i]$ is the prestress matrix and $[D_{Ti}]$ is the tangent material stiffness. As noted earlier, the solution involves either the use of constrained procedures[6] for appropriate load increment control or a direct Gaussian-type inversion scheme.[1]

To date such methodologies have been employed either in single processor or vector processor machines. The shortcomings of the FEM outlined in the previous section are essentially a direct outgrowth of the limitations of the architecture of single and vector-processor-type machines. In the next section, the intrinsic structure of the INR algorithm will be explored to define new computer architectures to bypass such difficulties.

5. Hierarchical Substructuring

From a conceptual point of view, the INR scheme defined by (2) does not confine the FEM scheme to a particular type of computer configuration. Rather the problems of speed and storage are essentially hardware based. Specifically the main questions and problems evolve out of the need to define architectures which enable the use of multiple processors so as to enhance overall machine speed as well as memory size. While the CRAY and CYBER systems are certainly a step in the right direction, they fall short of the ultimate requirements. Currently very large-scale FE models can easily outstrip the available core storage and machine CPU speeds.

In seeking to develop new computer architectures one is faced with the fact that

i) Vector processors require extensive cores as well as complex logic flows

ii) True parallel processors still await the fruition of properly organized DBM

Based on the foregoing, this paper seeks to develop what is called a hierarchical form of substructural parallelism. Following the pioneering efforts of the NASA Langley group[8,9] specifically, a nonlinear FE simulation, say of the tire, can be logically divided into a hierarchy of substructural groups defined by a variety of attributes, namely

i) Material group

ii) Geometric configuration

iii) Kinematic behavior

iv) Boundary conditions

At the lowest rung of the hierarchy, items (i)-(iv) are employed to define the specific local level substructural groups. The choice of the number of first-order groups is contingent on:

i) Minimizing core requirements of local level processors

ii) Minimizing number of perimeter nodes so that higher order substructural groups also have reduced core requirements for associated processors.

As can be seen, the main thrust is to maintain in core solutions for each local substructural processor.

Noting Figure 5, a given FE simulation can be broken up into a number of substructural levels. At each level internal nodes are eliminated to enable assembly through perimeter nodes. In terms of (2), the NR algorithm and its constrained counterpart can be substructured to yield the following first-level algorithms, that is:

$$\underset{\sim}{\Delta F}_{i+1}^{(1,k)} = [K_i^{(1,k)}]\ \underset{\sim}{\Delta Y}_{i+1}^{(1,k)} + \underset{\sim}{\Delta G}_{i+1}^{(1,k)} \tag{4}$$

k = 1,2,....Number of first-level substructure such that

$$\Delta \underset{\sim}{F}_{i+1}^{(1,k)} = \underset{\sim}{F}_{i+1}^{(1,k)} - \int_{R(1,k)} [B_i^*]^T \underset{\sim}{S}_i dv \tag{5}$$

$$\Delta \underset{\sim}{G}_{i+1}^{(1,k)} = \int_{R(1,k)} [N]^T \Delta \underset{\sim}{F}_{i+1} dv \tag{6}$$

where $(\ )^{(1,k)}$ denotes the first level k^{th} substructure, $(\ )_{i+1}$ the $(i+1)^{th}$ iteration, $\Delta \underset{\sim}{F}_{i+1}^{(1,k)}$ the nodal load increment, $[K_i]$ the substructural tangent stiffness, $\Delta \underset{\sim}{Y}_{i+1}^{(1,k)}$ the nodal deflection increment and $\Delta \underset{\sim}{G}_{i+1}^{(1,k)}$ the body force increment.

To enable assembly into second-order substructural groups, (4) is partitioned into internal and perimeter nodes yielding

$$\Delta \underset{\sim}{F}_{i+1}^{(1,k)} \rightarrow (\Delta \underset{\sim}{F}_{Pi+1}^{(1,k)} \quad \Delta \underset{\sim}{F}_{Ii+1}^{(1,k)}) \tag{7}$$

$$\Delta \underset{\sim}{Y}_{i+1}^{(1,k)} \rightarrow (\Delta \underset{\sim}{Y}_{Pi+1}^{(1,k)} \quad \Delta \underset{\sim}{Y}_{Ii+1}^{(1,k)}) \tag{8}$$

$$\Delta \underset{\sim}{G}_{i+1}^{(1,k)} \rightarrow (\Delta \underset{\sim}{G}_{Pi+1}^{(1,k)} \quad \Delta \underset{\sim}{G}_{Ii+1}^{(1,k)}) \tag{9}$$

$$[K_i^{(1,k)}] = \begin{bmatrix} [K_{PP}^{(1,k)}] & [K_{IP}^{(1,k)}] \\ [K_{IP}^{(1,k)}]^T & [K_{II}^{(1,k)}] \end{bmatrix} \tag{10}$$

Employing (7)-(10) we obtain the following relationships for the inner and perimeter nodes

$$\Delta \underset{\sim}{F}_{Pi+1}^{(1,k)} = [\kappa_{Pi}^{(1,k)}] \Delta \underset{\sim}{Y}_{Pi+1}^{(1,k)} + \Delta \underset{\sim}{f}_{Pi+1}^{(1,k)} \tag{11}$$

$$\Delta \underset{\sim}{Y}_{Ii+1}^{(1,k)} = -[\kappa_{PIi}^{(1,k)}] \Delta \underset{\sim}{Y}_{Pi+1}^{(1,k)} + \Delta \underset{\sim}{f}_{Ii+1}^{(1,k)} \tag{12}$$

where

$$[\kappa_{PPi}^{(1,k)}] = [K_{PPi}^{(1,k)}] - [K_{PIi}^{(1,k)}]^T [K_{IIi}^{(1,k)}]^{-1} [K_{PIi}^{(1,k)}] \tag{13}$$

$$\Delta f_{\sim Pi+1}^{(1,k)} = [K_{PIi}^{(1,k)}][K_{Ii}^{(1,k)}](\Delta F_{\sim Ii+1}^{(1,k)} - \Delta G_{\sim Ii+1}^{(1,k)}) + \Delta G_{\sim Pi+1}^{(1,k)} \tag{14}$$

$$[\kappa_{PIi}^{(1,k)}] = [K_{Ii}^{(1,k)}]^{-1} [K_{PIi}^{(1,k)}] \tag{15}$$

$$\Delta f_{\sim Ii+1}^{(1,k)} = [K_{IIi}^{(1,k)}]^{-1} (\Delta F_{\sim Ii+1}^{(1,k)} - \Delta G_{\sim Ii+1}^{(1,k)}) \tag{16}$$

Assembling (11) yields the second-level substructural relationships, namely

$$\Delta F_{\sim i+1}^{(2,k)} = [K_{i}^{(2,k)}] \Delta Y_{\sim i+1}^{(2,k)} + \Delta G_{\sim i+1}^{(2,k)} \tag{17}$$

k = 1,2,... Number of second level substructure

By partitioning (16) into inner and perimeter degrees of freedom we yield the third-order substructural relations after the appropriate assembly process. Continuing the partitioning and assembly process yields the various higher order substructural relations specifically

$$\Delta F_{\sim i+1}^{(j,k)} = [K_{i}^{(j,k)}] \Delta Y_{\sim i+1}^{(j,k)} + \Delta G_{\sim i+1}^{(j,k)} \tag{18}$$

wherein the associated inner and perimeter partitions take the form

$$\Delta F_{\sim Pi+1}^{(j,k)} = [\kappa_{Pi}^{(j,k)}] \Delta Y_{\sim Pi+1}^{(j,k)} + \Delta f_{\sim Pi+1}^{(j,k)} \tag{19}$$

$$\Delta Y_{\sim Ii+1}^{(j,k)} = -[\kappa_{PIi}^{(j,k)}] \Delta Y_{\sim Pi+1}^{(j,k)} + \Delta f_{\sim Ii+1}^{(j,k)} \tag{20}$$

such that

$$[\kappa_{Pi}^{(j,k)}] = [K_{PPi}^{(j,k)}] - [K_{PIi}^{(j,k)}]^T [K_{IIi}^{(j,k)}]^{-1} [K_{PIi}^{(j,k)}] \tag{21}$$

$$\Delta f_{\sim Pi+1}^{(j,k)} = [K_{PIi}^{(j,k)}][K_{IIi}^{(j,k)}](\Delta F_{\sim Ii+1}^{(j,k)} - \Delta G_{\sim Ii+1}^{(j,k)}) + \Delta G_{\sim Pi+1}^{(j,k)} \tag{22}$$

$$[\kappa_{PIi}^{(j,k)}] = [K_{IIi}^{(j,k)}]^{-1} [K_{PIi}^{(j,k)}] \tag{23}$$

$$\Delta f_{\sim Ii+1}^{(j,k)} = [K_{IIi}^{(j,k)}]^{-1} \Delta F_{\sim Ii+1}^{(j,k)} \tag{24}$$

Based on (11)-(24), we see that the overall nonlinear hierarchical substructuring requires a forward calculation phase as well as a backward stage. The forward phase involves the use of (11), (13), (14), (19), (21) and (22). In contrast the backward phase, which involves the definition of inner nodes, incorporates the use of (12), (15), (16), (20), (23) and (24). In terms of the forward iterative algorithms, the overall required machine architecture takes the form defined in Figure 6. Note the common data buses linking successive substructural levels need only provide access to perimeter data. In this way, significantly less data need to be accessed by the global-level DBM. This applies throughout the forward phase of the iteration process. Overall the steps handled by each of the succeeding levels involve assembly, inner/perimeter partitioning, and setting up effective stiffnesses for the forward and backward phases. In terms of (21)-(24), the stiffnesses associated with the perimeter and inner nodes involve an inverse of the inner partition of the k^{th} substructural stiffness. All such manipulations must be performed by processors dedicated to each of the k individual substructures associated with the various hierarchical levels.

Once the forward loop of calculations is complete, the perimeter data must be back tracked to the inner nodes of each of the various substructures at the different substructural levels. The overall flow of control/calculation is depicted in Figure 7. As can be seen, the perimeter data are used to determine the inner nodal incremental excursions. This is achieved through the use of the family of expressions defined by (20). Once the back substitutions to the succeeding levels up to and including the first are completed, the standard norm type convergence checks must be implemented to ascertain the quality of convergence. Contingent on the convergence check, the iteration process can be cycled through the forward and backward phases of the substructural hierarchy.

6. Discussion

To illustrate the hierarchical substructural scheme, consider the three-level simulation defined in Figure 8. The number of nodes and substructure associated with the example are given in Figure 9. Based on the number of inner and perimeter variables depicted, the expressions defining the number of respective nodes are given by:

i) Level 1

$$\text{Perimeter Nodes} = 2(\ell_1 + \ell_2 + \ell_3) \tag{25}$$

$$\text{Inner Nodes} = (\ell_1 - 2)(\ell_2 - 2) \tag{26}$$

ii) Level 2

$$\text{Perimeter Nodes} = 2(\ell_1 n_1 + \ell_2 n_2 - n_1 - n_2) \tag{27}$$

$$\text{Inner Nodes} = n_1 n_2(\ell_1 + \ell_2) - n_1(\ell_1 + 2) - n_2(\ell_2 + 2) - 3n_1 n_2 + 1 \tag{28}$$

iii) Level 3

$$\text{Perimeter Nodes} = 2m_1 n_1(\ell_1 - 1) + 2m_2 n_2(\ell_2 - 1) \tag{29}$$

$$\text{Inner Nodes} = \ell_1 m_1 n_1(m_2 - 1) + \ell_2 n_2 m_2(m_1 - 1) - m_1 m_2(n_1 + n_2) + m_1 n_1 + m_2 n_2 - m_1 m_2 + 1 \tag{30}$$

Employing (25-30) we see that the storage effectiveness of each of the various levels is expressed by the relations

$$\xi^{(1)} = \frac{\text{Perimeter}}{\text{Perimeter + Inner}} \tag{31}$$

where k denotes the level number. In the context of (31), it follows that

$$\xi^{(1)} = \frac{2(\ell_1 + \ell_2 - 2)}{\ell_1 \ell_2} \tag{32}$$

$$\xi^{(2)} = \frac{2(\ell_1 n_1 + \ell_2 n_2 - n_1 - n_2)}{n_1 \ell_1(n_2 + 1) + \ell_2 n_2(n_1 + 1) - 3n_1 n_2 - 4(n_1 + n_2) + 1} \tag{33}$$

$$\xi^{(3)} = \frac{2m_1 n_1(\ell_1 - 1) + 2m_2 n_2(\ell_2 - 1)}{\ell_1 m_1 n_1(m_2 + 1) + \ell_2 m_2 n_2(m_1 + 1) - m_1 m_2(n_1 + n_2 + 1) - m_1 n_1 - m_2 n_2 + 1} \tag{34}$$

Consider the case wherein

$$\ell_1 = 100, \ \ell_2 = 50$$

$$n_1 = 5, \ \ n_2 = 4$$

$$m_1 = 3, \ \ m_2 = 4$$

In terms of the foregoing, Table 2 gives the total number of

- degrees of freedom
- processors required at each level
- perimeter/inner nodes

as well as the storage effectiveness of each of the substructural levels. Noting that a straight solution of the given problem would require a 1.2×10^6 order stiffness matrix, it follows from Table 2 that very significant storage savings as well as speed enhancements can be achieved.

In the context of the foregong development, it follows that hierarchical substructural parallelism has decided advantages over vector-type processors, namely:

i) Global common core is reduced in size

ii) Substructures are handled in smaller local cores which could employ vector processors and which are controlled by local DBM

iii) Data transfer between succeeding levels of substructural hierarchy are reduced thereby reducing load on DBM

iv) Various substructures are updated, inverted, and assembled simultaneously hence emhancing the overall speed

v) The overall addressing requirements are reduced since the size of individual substructural zones is much smaller

iv) Extensive use of cash memory (Ram Disk) can be made at the local level thereby reducing disk I/O

vii) Backward and forward steps follow natural formulational lines

viii) Element-to-element or hierarchical least-squares algorithms can be employed at the local substructural level

ix) Linear/nonlinear problem partitioning can be more logically handled

x) Overall control of the machine is more logical and less difficult since local processors are essentially autonomous within updating and inverting phases of the operation

xi) The MCL can be patterned about well-defined substructuring methodology; the transfer of control from level to level is contingent on the monitoring status of stiffness/inversion calculations

xii) The data base manager needs only to deal with data residing on perimeters of the substructure; as noted earlier, this significantly reduces the amount of data transferred between levels.

As discussed earlier, the modeling of tires in their use environment represents perhaps one of the most comprehensive single component nonlinear structural response problems currently available. This follows from the fact that geometric-, material-, and boundary-induced nonlinearity all simultaneously act to define the global response behavior. Due to their regionalized/ substructural form of construction, tires represent a good modeling problem to help define the architecture of high-level multiprocessor machines. In this context, a hierarchical form of substructural parallelism has decided advantages over other forms of multiprocessors. As has been seen such a procedure has several theoretical advantages for nonlinear problems. These evolve about the simplified DBM structure, reduced data flow, smaller global core, and reduced addressing requirements.

Overall future work in this area should

- Place qreater emphasis on algorithmic architecture and its possible effects on machine structure
- Establish proper control configuration for hierarchical DBM
- Extend scheme to constrained incremental Newton/Raphson (INR) least-squares algorithms as well as transient schemes
- Apply concept to available parallel processors
- Structure procedure so as to enable either direct or iterative solutions at substructural level
- Establish criteria to enable determination of quality of convergence at local substructural level

6. References

1. Zienkiewicz, O.C., The Finite Element Method, McGraw Hill Company (1983).

2. Bathe, K.J., Finite Element Procedures in Engineering Analysis, Prentice Hall (1983).

3. Oden, J.T., Finite Elements of Nonlinear Continuum, McGraw Hill Co. (1972).

4. Cook, R.D., Concepts and Applications of Finite Element Analysis, Wiley (1983).

5. Hughes, T.J.R., Winget, J. and Levit, I., Comp. Meth. Appl. Mech. Engineering, 36, 241 (1983).

6. Padovan, J. and Arechaga, T., Int. Jr. Engineering Science, 20, 1077 (1982).

7. Padovan, J. and Lackney, J., Computers and Structures, 19, 535 (1984).

8. Noor, A.K., Storaasli, O.O. and Fulton, R.E., ASME Special Publ., H00275, 1 (1983).

9. Bostic, S.W. and Fulton, R.D., AIAA Paper No. 85-0783-CP (1985).

10. Padovan, J. and Tovichakchaikul, S., Computers and Structures, 18, 191 (1984).

11. Padovan, J., Global-Local Finite Element Modeling, 126th Meeting of the ACS Rubber Division (1984).

12. Clark, S., Mechanics of Pneumatic Tires, U.S. Department of Transportation (1980).

13. Padovan, J. and Nakajima, Y., Numerical Simulation of Tire Skidding Events Involving Impacts with Holes and Bumps, 4th Annual Tire Society Meeting (1985).

14. Padovan, J. and Nakajima, Y., Numerical Simulation of Tire Transiently Sliding/Rolling Over Holes and Bumps, ACS Meeting (April 1985).

15. Padovan, J. and Paramadilok, O., Transient and Steady State Viscoelastic Rolling Contact, Computers and Structures, 20, 545-553 (1985).

16. Trinko, J.J., Ply and Rubber Stresses and Contact Forces for a Loaded Radial Tire, First Annual Meeting of Tire Society, Akron (1982).

17. Tabaddor, F. and Stafford, J., Computers and Structures, 13, 737 (1981).

TABLE 1 COMPARISON OF LINEAR AND NONLINEAR FE SIMULATION OF PRESSURIZED TIRE

PRESSURE (PSI)	LINEAR (IN) MAXIMUM DEFLECTION		NONLINEAR (IN) MAXIMUM DEFLECTION	
	CROWN	SIDEWALL	CROWN	SIDEWALL
5.	.003	.036	.003	.027
10.	.006	.073	.006	.047
15.	.009	.109	.009	.064
30.	.018	.218	.018	.107

TABLE 2 COMPARISON OF HIERARCHICAL SUBSTRUCTURAL PARALLEL AND SINGLE PROCESSOR SYSTEMS

LEVEL	HIERARCHICAL SUBSTRUCTURAL PARALLEL SYSTEM					SINGLE PROCESSOR
	NUMBER OF PROCESSORS	PERIMETER D of F/P*	INNER D of F/P	TOTAL D of F/P**	STORAGE EFF.	D of F
1	240	296	4704	5000	.0592	—
2	20	1382	2223	4605	.3	—
3	12	4538	6012	10550	.3	—
GLOBAL	—	—	—	—	—	1200000

*D of F/P - degrees of freedom per processor

**Total number of perimeter and inner D of F

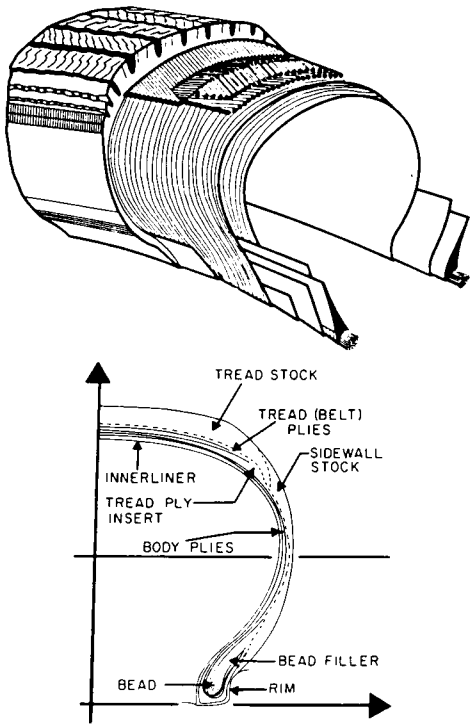

Figure 1 Tire geometry and cross section.

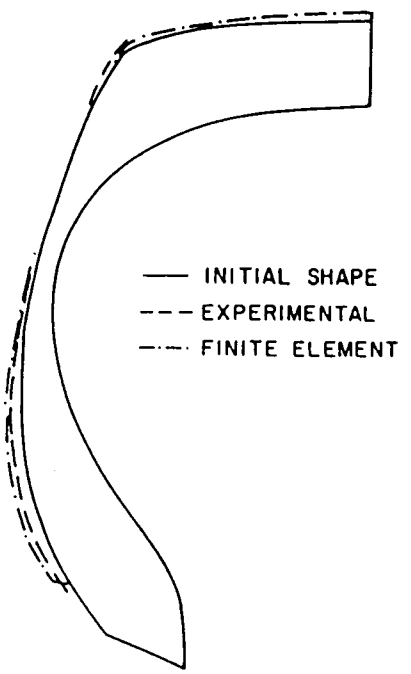

Figure 2 Pressure tire profile.

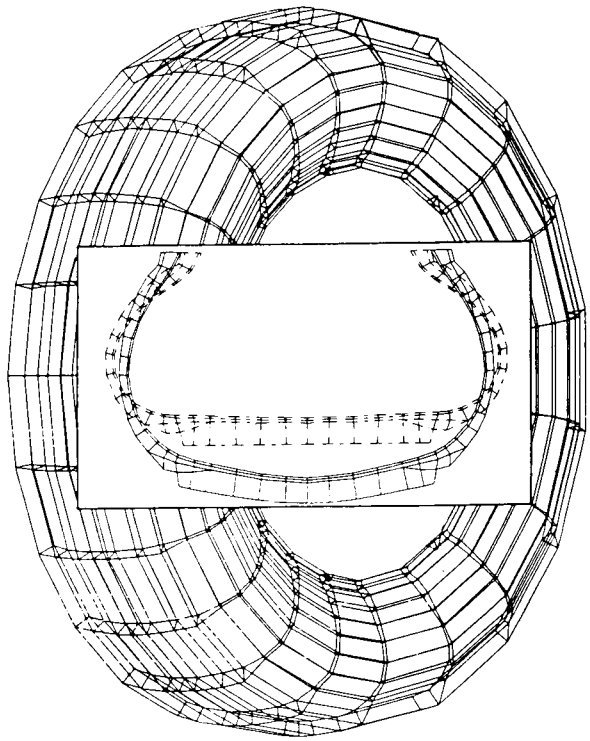

Figure 3 Loading into standing contact.

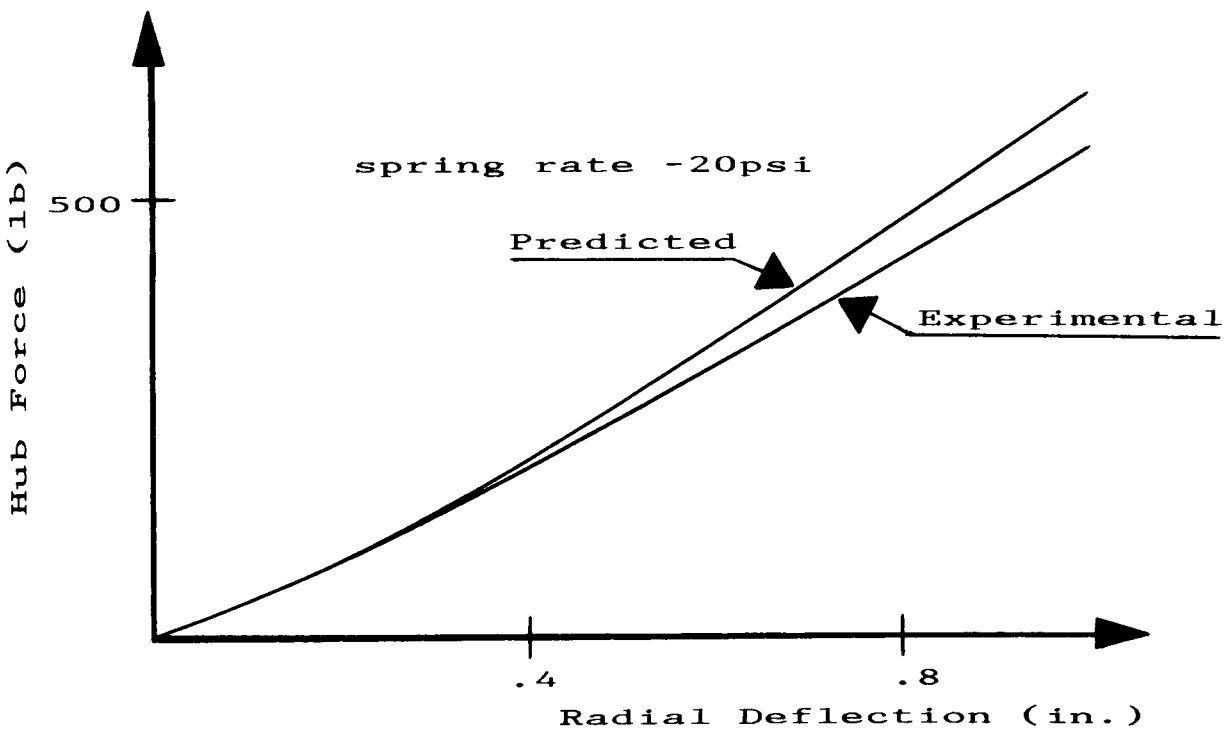

Figure 4 Force deflection characteristics into standing contact.

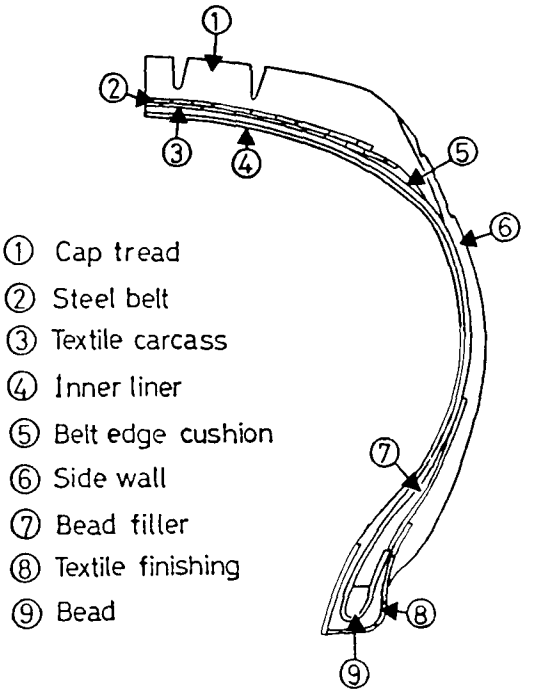

Figure 5 Substructural zones of tire.

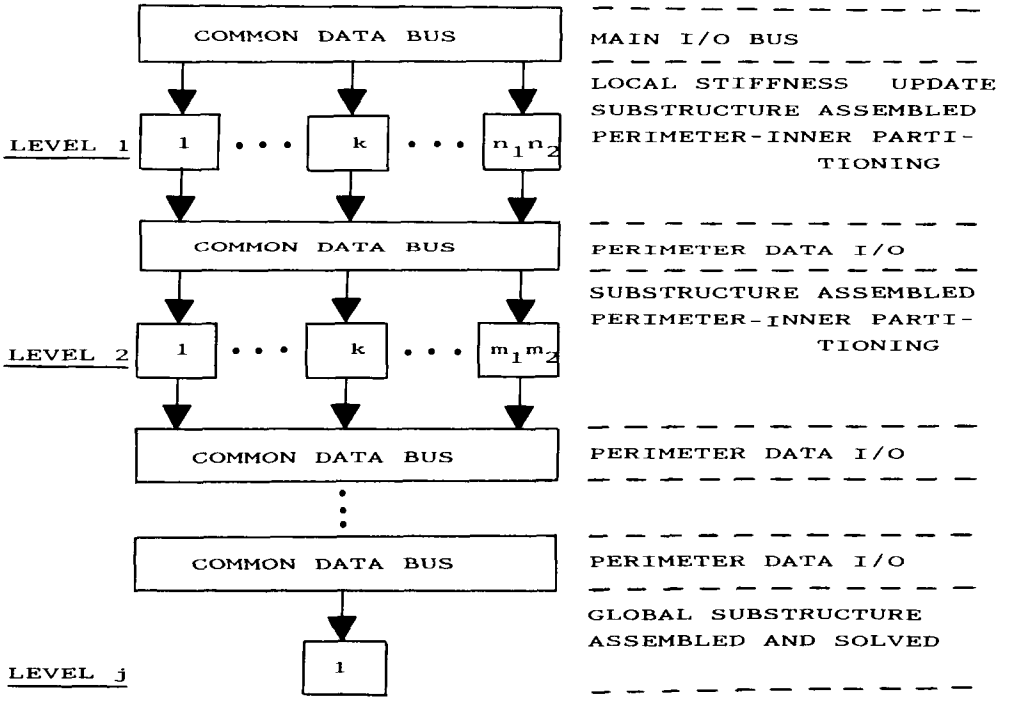

Figure 6 Flow of control: forward loop.

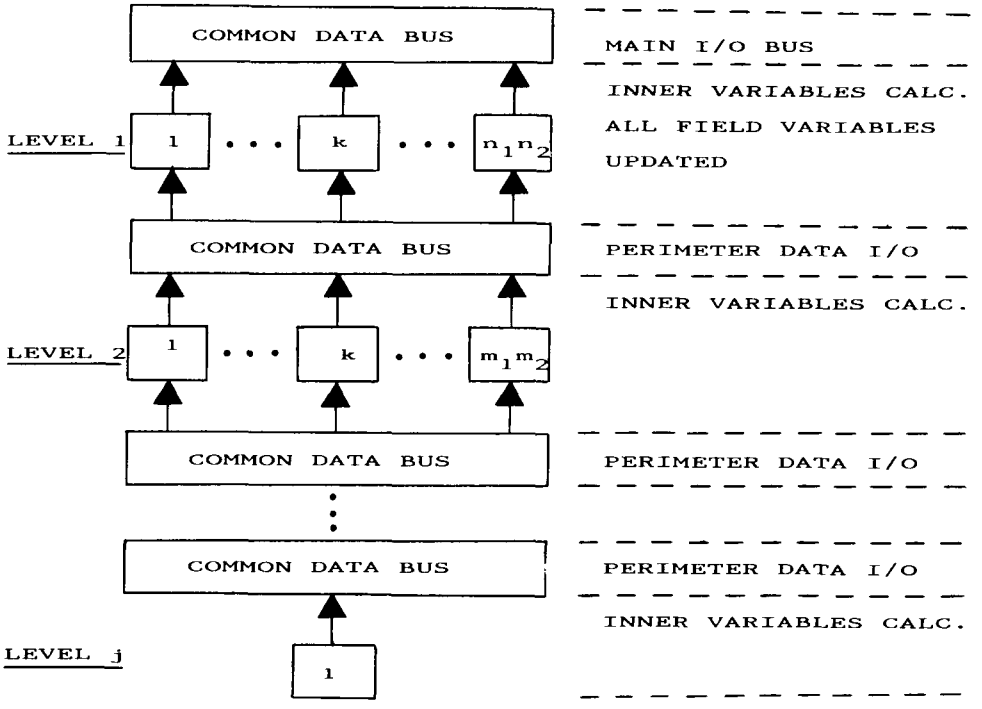

Figure 7 Flow of control: backward loop.

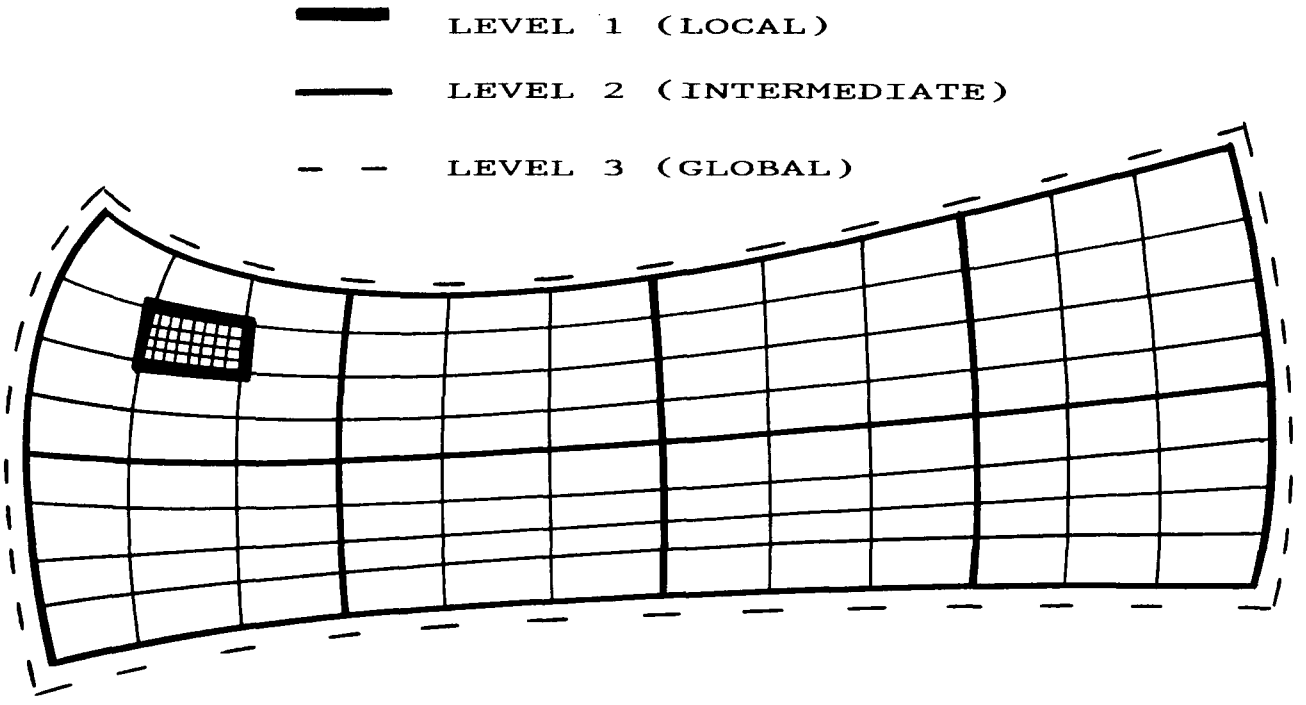

Figure 8 Example of three-level hierarchical substructural system.

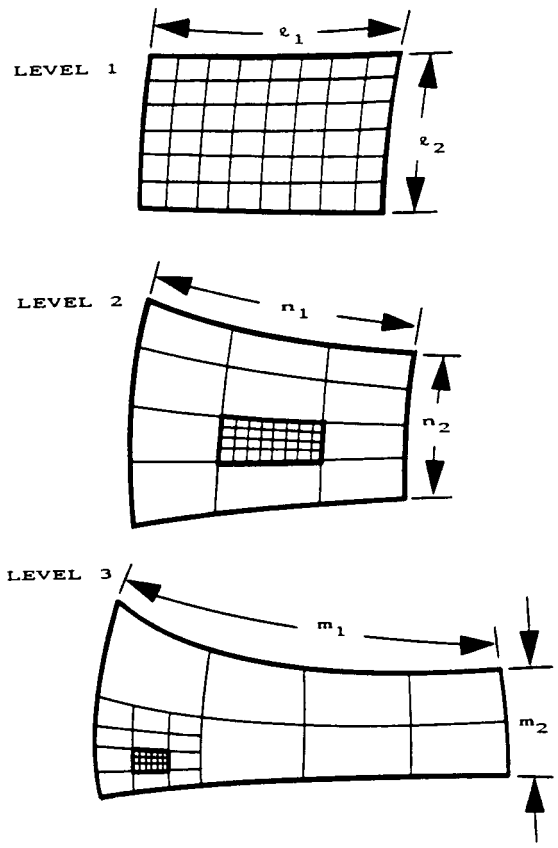

Figure 9 Number of perimeter substructures at each level.